CENTRE FOR
SYSTEMS
SCIENCE

LCCR

Simon Fraser University, Burnaby, B.C., Canada V5A 1S6 / (604) 291-4704

# Head-Driven Phrase Structure Grammar as an Inheritance Hierarchy

Carl Vogel and Fred Popowich

CSS/LCCR TR90-03
CMPT TR90-03

April 1990

92-05703/

# Head-Driven Phrase Structure Grammar as an Inheritance Hierarchy

**Carl Vogel and Fred Popowich**

CSS/LCCR TR 90-03
CMPT TR 90-03

April 1990

Centre for Systems Science
School of Computing Science
Simon Fraser University
Burnaby, British Columbia
CANADA V5A 1S6

vogel@cs.sfu.ca
Fred_Popowich@SFU.BITNET

**Abstract**

An instance of a cyclic inheritance network is presented. The network captures most of the conceptual structure of Head-Driven Phrase Structure Grammar (HPSG) (Pollard and Sag, 1987). This paper describes the content of nodes (information structures encoded as attribute-value matrices) and the semantics of links (the relative psuedocomplement relation) in this inheritance hierarchy. Further, the paper specifies the nature of the path-based reasoning required by HPSG over the network, reasoning which is not confounded by the presence of a cycle. The implications of this work for theorists in both unification grammars and inheritance reasoners are discussed.

## Table of Contents

1. Introduction: HPSG is an Inheritance Network 1
2. Nodes 1
3. Links 2
4. The Inheritance Network for HPSG 4
5. Inheritance Reasoning 5
6. An Example 6
7. Applications of Inheritance Reasoning for HPSG 8
8. Conclusions 9
Acknowledgements 10
References 10

## 1. Introduction: HPSG is an Inheritance Network

Head-Driven Phrase Structure Grammar (HPSG) is in the family of linguistic theories known as unification grammars (Shieber, 1986). Two aspects of HPSG are of interest to us: one is its nature as a formal language for expressing a theory of linguistics (i.e., for specifying grammar rules, lexical entries, and universal principles), and the other involves a use of that formal language to describe a specific grammar which actually makes predictions about linguistic phenomena. Both of these views can be captured with in the framework of inheritance reasoning systems. That is, we argue that HPSG is an instance of an inheritance network, and we demonstrate how both views of HPSG can be obtained through specific sorts of processing over the hierarchy. We describe the peculiar demands that the network imposes on inheritance reasoning systems to yield appropriate results. And, finally, we indicate the relationship between inheritance reasoning and parsing.

HPSG is a multiple inheritance hierarchy in two dimensions. The first dimension is definitional. This *primitive* dimension facilitates the specification of HPSG as a formal language. The exact specifications are outlined in (Pollard and Sag, 1987). Pollard and Sag also include discussion of the inheritance which is at work here. The mode of reasoning executed over these networks, at least in the hierarchy of lexical types, is a combination of shortest path and skeptical reasoning[1]. We are concerned in the present work with the second dimension, the *operative* hierarchy. The first dimension is used to articulate the structure of t-signs[2], but the operative dimension relates t-signs to each other.

## 2. Nodes

T-signs are the formal constructs that we use to encode the information relevant to linguistic phenomena. HPSG is an interesting formal language because it uses t-signs both to encode information abstracted from lexical items, phrases, and sentences, and to express constraints on the combination (or decomposition) of t-signs. Roughly, to be the t-sign for a specific sentence, the putative t-sign must *unify* with the t-sign that expresses the constraints sanctioned by the theory. It is not as simple as this, though, since some of the constraints are expressed through relations between t-signs stated in terms of the relative pseudocomplement operator (=>).

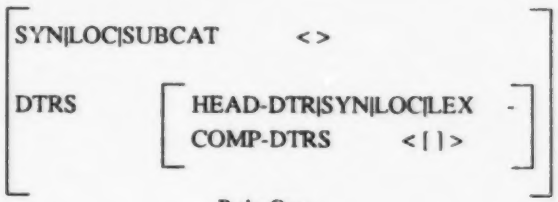
We take t-signs to correspond to nodes in an inheritance hierarchy. This is

---

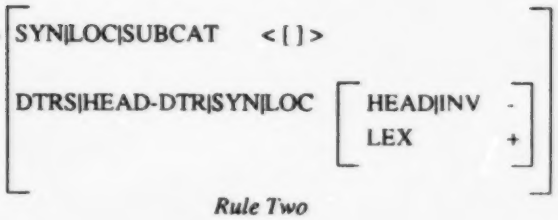
[1]See, for instance, (Shieber, 1986) who speculates on the skeptical behavior in the face of inconsistencies, as well as (Flickinger, et.al., 1985) who describe two corresponding modes called *normal* and *complete* inheritance.

[2]Pollard and Sag actually call the theoretical construct *signs*, but we chose a distinct name for the same object to emphasize the fact that we are talking about an object in a formal language. Thus, we avoid all debate about the ontological status of signs.

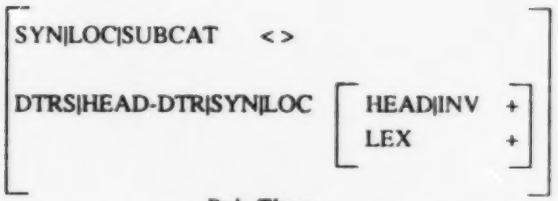
reasonable, after all, since t-signs are attribute-value matrices; they are information structures. Inheritance networks purport to represent concept hierarchies, and we interpret t-signs as formal encodings of concepts[3]. More precisely, the nodes in the operative hierarchy of HPSG are the t-signs which encode the grammar's rules and principles. For example, three of the nodes contain the information given in the t-signs of Figure 1, one t-sign at each node.




*Rule One*




*Rule Two*



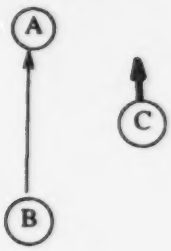

*Rule Three*

**Figure 1:** Grammar Rules.

## 3. Links

HPSG includes several universal grammar principles which are stated using =>. When this operator is applied to information structures A and B as in, "A => B," it yields an information structure, C, the least one whose unification with A is subsumed by B (Pollard and Sag, 1987, p.43). If A is at least as informed as B, then C is vacuous (it will unify with anything). If B is more informed than A, then C is the information in B that is not in A. In the corresponding inheritance network, C is the information that is inherited. The arrowheads on links in Figure 2 indicate the direction of information flow. In Figure 2, when it is the case that B contains more information than A, C is the information that

[3]Kasper and Rounds go so far as to ground these encodings in formal logic (Kasper and Rounds, 1986).




A => B

**Figure 2:** Relative Pseudocomplement and the Operative Hierarchy.

is inherited.

If it happens that A and B are inconsistent with respect to each other (A and B fail to unify; A ∧ B yields *bottom*), then the information inherited through C to A causes an inconsistency at that node. Further chaining of links is fruitless, because *bottom* was inherted to A, and *bottom* will be inconsistent with any other node linked to B through A. This situation corresponds to the presence of a negative link in the hierarchy. Given a procedural interpretation, this means that the polarity of a link in the hierarchy with the semantics of => is determined dynamically, depending on the information contained in nodes at either end. Declaratively, we can say that a link is used in a network to represent a relationship between two nodes; the content of the link, its polarity and the information inherited across it, is determined in terms of the stated relation by the information contained at the respective nodes.

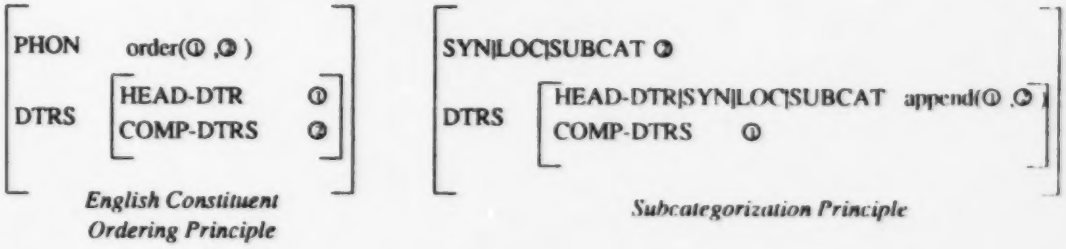



**Figure 3:** Universal Grammar Principles.

In presenting the universal principles of grammar, Pollard and Sag use the relative pseudocomplement operator, and in their usage A is vacuous with respect to a class of t-signs (headed structures, for instance) — that is, A conveys no information except for the class of t-signs used as a reference set, but B is also assumed to be a member of this set. This forces the inheritance of all the information contained in B just as in classical

implication. This allows us to present the grammar principles as individual t-signs rather than as relations between t-signs: the t-signs in Figure 3 show the information that is actually inherited in a couple of the universal grammar principles.

## 4. The Inheritance Network for HPSG

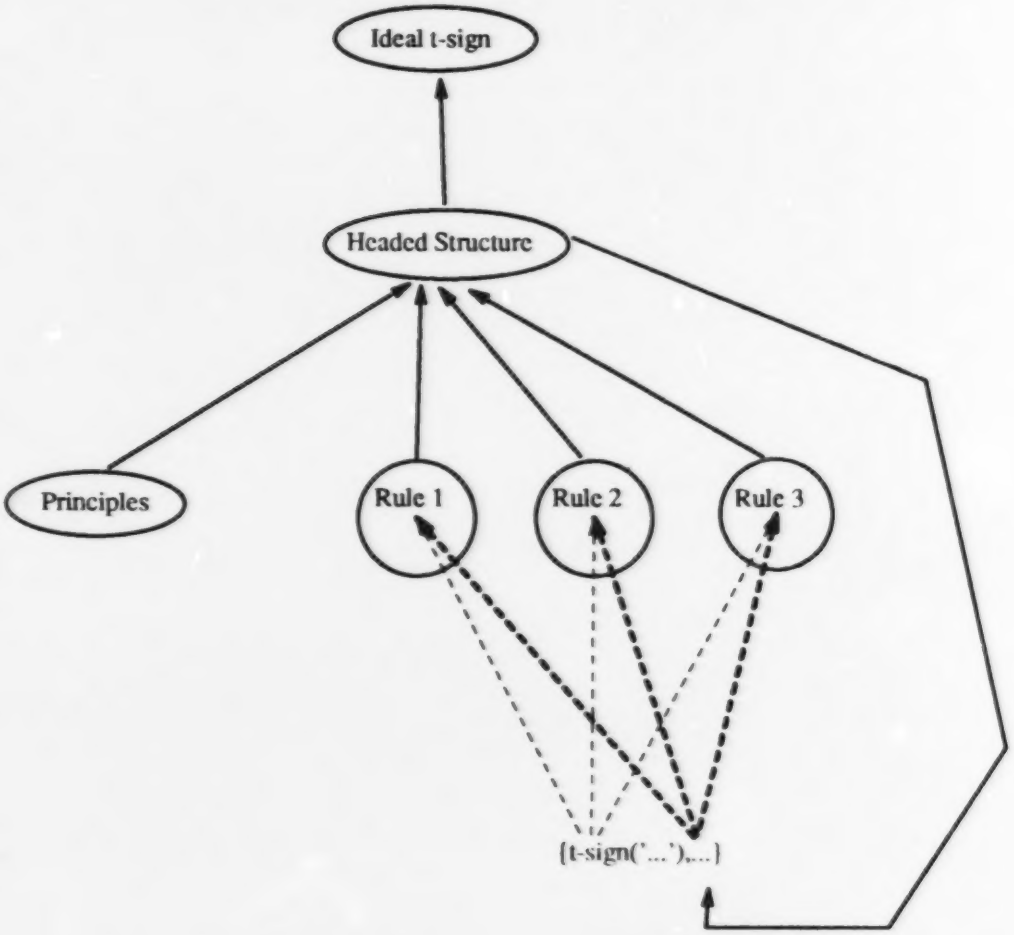



**Figure 4:** The Operative Dimension of HPSG as an Inheritance Hierarchy.

Finally, having described the content of nodes and the semantics of links, we are in a position to give the inheritance network which corresponds to the operative dimension of HPSG[4]. We see from Figure 4 that there are several departures from the usual notation for inheritance networks. First we note that there are three sorts of arrows. The dashed

[4]We have presented the inheritance hierarchy for only that subset of HPSG whose implementation was described in (Popowich and Vogel, 1990); however, this subset constitutes most of HPSG, and we suspect that when analyzed, the remainder of HPSG will be consistent with the treatment described herein, as well.

arrows are defeasible. This indicates that any given rule may or may not apply to the set of t-signs below it. A dark dashed arrow has the semantics of an operator called ->, also a relative pseudocomplement operator, but one which is a generalization of => over lists of t-signs (i.e., it relates a list of t-signs to another list of t-signs). We also note that some of the nodes, the ones labeled for rules, have arrows pointing into them rather than simply to them. The is a useful notion which exploits the fact that inheritance sometimes involves complex concepts at nodes. Essentially, we allow inheritance to a node through component concepts of the node. With respect to the t-signs presented in Figure 1, the list-valued links are for inheritance into the COMP-DTRS attribute, and the defeasible links point to the HEAD-DTRS attribute.

Other aspects of the hierarchy will become apparent as we explain how we reason over the network. This includes an explanation of the nodes at the top and bottom of the hierarchy, as well as the link which renders the network cyclic[5]. Just as in the inheritance literature (c.f., Touretzky (1986), Touretzky, et.al. (1987)), reasoning is based upon the construction of valid paths through the network.

## 5. Inheritance Reasoning

We have the operative hierarchy which emerges from our language (HPSG) for describing linguistic phenomena. We begin with a sentence and a set of features that we think are relevant and inherent in the structure of that sentence, so we know how we want to classify the sentence using the language of t-signs. This classification is the "ideal" t-sign which is at the top of the hierarchy. In the language of t-signs, we also have proposals about the way t-signs should interact. These are the principles and nodes which constitute the lower nodes in the hierarchy. The bottom node is a partially specified set of t-signs which can be constituents of the ideal. We use the operative hierarchy to verify that it is possible to trace an inheritance path through the network, from the set of constituent structures at the bottom, through all of the composite constituent structures, to the structure of the ideal. If a path exists, then the ideal t-sign is also a member of the set at the bottom. If more than one path is found, we want to keep all of them (because they all represent valid, if distinct, analyses). That we keep all valid paths, even if they do not agree with each other, indicates that we sanction a form of credulous reasoning — we do not refrain from reaching conclusions when several mutually inconsistent conclusions are available. If a path is not found, it means, by virtue of =>, that an inconsistency has been inherited.

Any particular *use* of the operative hierarchy is an exercise in determining the analysis (or, for that matter, the generation) of an utterance expressed in t-signs, as

---

[5]One restriction imposed by inheritance reasoners is that the graphs are assumed to be acyclic. Clearly, the reasoning which we anticipate against our network must be restricted in some sense to accommodate cyclicity.

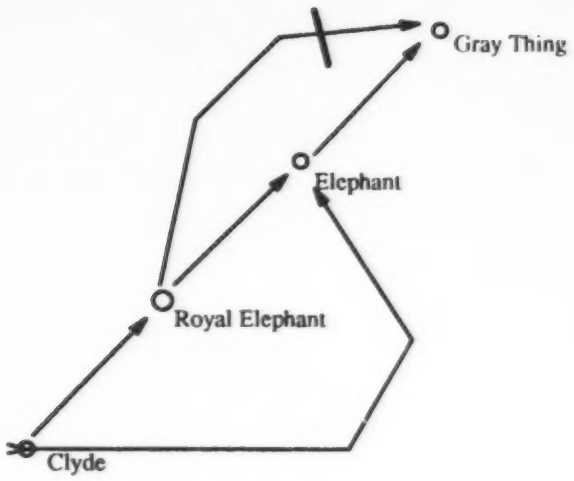



**Figure 5:** Touretzky's Example of a Concept Hierarchy.

sanctioned by an HPSG grammar. So, to demonstrate the form of inheritance reasoning we require, we provide an HPSG analysis of a simple sentence, "Mary walks," using the operative hierarchy. The literature on inheritance reasoning systems provides example hierarchies like the one in Figure 5 and shows the process of reasoning about its nodes, as in, for instance, determining whether Clyde is a gray thing. So, too, for HPSG: we begin with the operative hierarchy and a sentence, "Mary walks", and ascertain whether t-sign("Mary walks")[6], the node at the top of the hierarchy, is a member of the initially undetermined part of the set of t-signs at the bottom of the hierarchy, {t-sign("Mary"),t-sign("walks"), ...}. If it is not a member then inheritance in the hierarchy yields an inconsistency, and this means that the sentence with which we began is not sanctioned by HPSG as a grammatical utterance.

## 6. An Example

Using this example we have an initially undetermined set, {t-sign("Mary"), t-sign("walks"),...}. Consider a member of this set, t-sign("walks"), for instance in Figure 6, and recall the t-sign for Rule One, given in Figure 1. T-sign("walks") fails to unify with the HEAD-DTR of Rule One, because the SYNILOCILEX feature of t-sign("walks") is specified to be "+". Therefore, we have an inconsistency, a negative link in the hierarchy, and we proceed no further *on that* path (we follow the convention that there can be at most one negative link in a path, and if a negative link exists in a path, it is the

[6]We allow ourselves the notational ease of selecting the t-sign relevant to discussion, whether it be from the set of ideals, for complex sentences, or from the lexicon for specific words, by writing it as a function of the appropriate string.

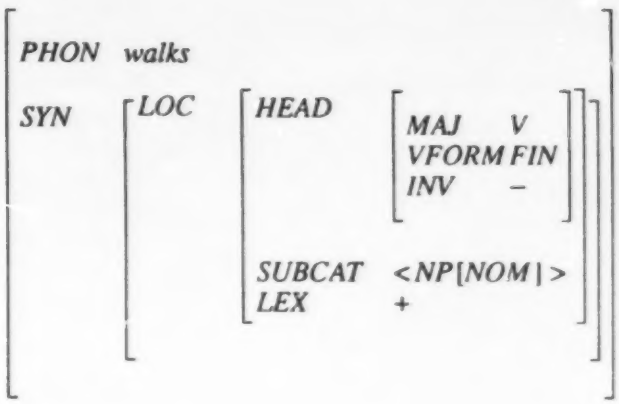



**Figure 6:** Lexical Entry for "walks".

last link in the path — recall the earlier mention of futility of chaining past a negative link). But, since that initial link is defeasible, we are still able to consider other paths. The path through the node for Rule Three is also a negative link because of the specification of the SYN|LOC|HEAD|INV feature for t-sign("walks"), and that leaves the node for Rule Two. The unification of t-sign("walks") with HEAD-DTR of Rule Two succeeds (though, it is important to note that we do *not* perform a destructive unification); the value of the COMP-DTRS feature of Rule Two unifies with the empty list. We consider the inheritance of information from the *result* of that unification with the node for the grammar principles' t-sign (only one t-sign, since we constructed from the t-sign for each principle, the single t-sign which represents their unification), for that information inherits to the "Headed Structure" node, as well. This unification also succeeds, and though unification with t-sign("Mary walks") does not, inheritance is admitted along the path to the set of t-signs at the bottom of the hierarchy, completing a cycle, and further specifying that set — the t-sign in Figure 7 is known to be a member.

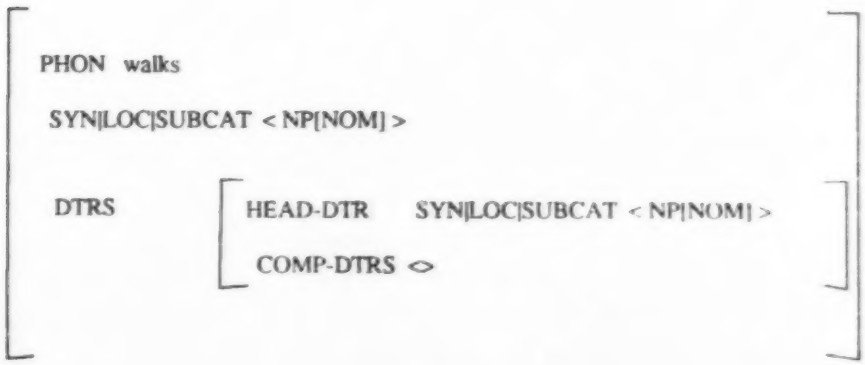



**Figure 7:** Inherited Information.

This t-sign will unify with the HEAD-DTR of Rule One (note that it will also unify with the HEAD-DTR of Rule Two since, as mentioned above, the earlier unification was not destructive) since it is not specified for the SYN|LOC|LEX feature at all, and furthermore, <t-sign("Mary")> will unify with the value of COMP-DTRS on Rule One. The result of these unifications is itself consistent with the node for the principles, hence inheritance is successful to that node. It also happens that t-sign("Mary walks") is consistent with the result from the last point, which, as before, is also inherited back down to the set of t-signs that is not fully determined, and this implies that t-sign("Mary walks") is indeed a member of that set. Hence, what would be considered the "ideal" feature specification of the sentence, "Mary walks," using the primitive hierarchy and the intuitions that the linguists had hoped to capture is actually realized in the HPSG's operative dimension. The sentence, "Mary walks," is considered grammatical. Formally speaking, it is grammatical because at least one path exists from a partially specified set of t-signs which includes the lexical entries for each word in the sentence, through the concept nodes outlined by the theory, to the ideal t-sign specification. The presence of more than one such path would indicate an ambiguity. An utterance is ruled ungrammatical only if no path is permitted through the hierarchy. Here, the cyclic structure does not pose the problems that it would for inheritance systems in general, because the nature of this aspect of the operative hierarchy's employment did not require the resolution of conflicting paths. For this reason, neither was the presence of redundant paths a problem, because multiple paths represent multiple analyses. We do not want to disregard any of them.

## 7. Applications of Inheritance Reasoning for HPSG

Recall that there are two aspects of HPSG, grammars written within the formalism, and the formalism itself. Both aspects involve the operative hierarchy, but the distinction is apparent with respect to the sentences we pose against the hierarchy. To use it as a grammar formalism is to use HPSG as a formal language to specify a grammar, which of course may require debugging, so we pose simple structures whose ideal t-signs are obvious, and we verify that inheritance is permitted from constituent structures. On the other hand, when we use a particular HPSG grammar, we assume that it does not need any adjustment, and we use it to construct t-signs to discover what they look like and examine its predictions in general for novel sentences.

The example described above, for "Mary walks," demonstrates the use of HPSG's operative hierarchy as a grammar formalism. We began with simple sentences, whose complete t-signs are apparent, and we verified that the structure of the grammar actually allows us to construct ideal t-signs from the t-signs which should be their constituent structures, according to the various principles. If inheritance is not allowed, then we know that the grammar has to be adjusted, either in the way information is abstracted into t-signs in the lexicon or in the way information and structures are constrained to combine

by the principles and rules. It is also possible further consideration will deem that some paths through the network reflect spurious analyses which need to be ruled out by changing the nodes in the hierarchy[7].

Clearly, the second use of the operative hierarchy is closely related to parsing. When we parse a sentence we determine its constituent analysis just as in the second use of the operative hierarchy we determine a t-sign formulation for a sentence (if one exists). So, a machine implementation of an HPSG grammar is a machine implementation of a reasoning mechanism which operates over the HPSG operative hierarchy. Of course, it is valid (and a claim that we in fact make when we say that we have constructed a *tool* which is useful to linguists) to point out that a parser is useful for the grammar verification usage of the hierarchy as well, since if a t-sign cannot be built for a sentence that should be covered (or if the parser constructs a t-sign for a sentence that it should not), then we have used the parser to discover errors in the grammar's predictions, and a problem exists with the grammar that needs to be patched. But, in the abstract, the problem is patched by determining the structure that should be present, the nature of the ideal t-sign, and how the nodes in the hierarchy need be affected in order for inheritance to go through.

## 8. Conclusions

It is interesting to examine the operative dimension of HPSG as an instance of inheritance reasoning because it allows us to comment on the applicability of formal systems for inheritance reasoning. The network we distilled from HPSG is a nontrivial concept hierarchy (at each node except for the node for the list of individuals at the bottom we have t-signs, and using unification we can see a single t-sign as an abbreviation for a class of t-signs, so our concepts are identified by the t-signs which serve as the abbreviations for larger classes). Moreover the t-signs exist in a subsumption ordering which happens to correspond largely to taxonomic orderings. This all means that we have a real live example of a taxonomic arrangement of concepts about which we wish to reason — we can potentially test out the intuitions and usefulness of various inheritance schemes on an actual hierarchy of concepts without having to construct artificial examples. Further work along this line will construct similar networks as appropriate to other unification grammar formalisms (for instance, Tree Unification Grammar, (Popowich, 1989) and Unification Categorial Grammar, (Calder, Klein, and Zeevat, 1988)), with in mind the dual goal of exploring the implications of complexity results for inheritance reasoners (c.f. Selman and Levesque (1989)) on parsing techniques and of testing out the generality of the relative pseudocomplement as defining the semantics of links in an arbitrary inheritance network; the relationship between the

[7]This occurred in our research (Popowich and Vogel, 1990) leading us to a refinement in the statement of Rule Two.

semantics suggested here and that suggested by other researchers (c.f. Pearl (1987), Boutilier (1989), Krishnaprasad (1989), Delgrande (1990), et.al.) demands further investigation, as well.

We find the work presented in this paper to be another step in a general trend (along with increased lexicalization of computational grammar systems) in tying parsing paradigms to knowledge representation frameworks. In fact, the analysis provides a pedagogical tool for explaining the operative dimensions of the grammar formalism through an analog to knowledge representation schemes. We are excited by the depth of the connections which exist between these two seemingly disparate areas -- unification grammar parsing and inheritance reasoning — and we are confident that further study will result in a fruitful exchange of ideas and techniques between both areas.

## Acknowledgements

The research presented in this paper is supported by the Centre for Systems Science of Simon Fraser University, the Advanced Systems Institute of British Columbia, and the Natural Sciences and Engineering Research Council of Canada under Grant Nos. OGP0041910 and A4309.

## References

Boutilier, Craig. (1989). A Semantical Approach to Stable Inheritance Reasoning. *Proceedings of the 11th International Joint Conference on Artificial Intelligence*. Detroit, Michigan.

Calder, Jo, Ewan Klein, and Henk Zeevat. (1988). Unification Categorial Grammar: A Concise, Extendable Grammar for Natural Language Processing. *12th International Conference on Computational Linguistics*. Budapest, Hungary.

Delgrande, Jim. (1990). A Semantics for a Class of Inheritance Networks. *Proceedings of the 8th Biennial Conference of the Canadian Society for Computational Studies of Intelligence*. Ottawa, Ontario.

Flickinger, Dan, Carl Pollard and Tom Wasow. (1985). Structure-Sharing in Lexical Representation. *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*. University of Chicago, Chicago, IL.

Kasper, Robert, and William Rounds. (1986). A Logical Semantics for Feature Structures. *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics*. Columbia University, New York, NY.

Krishnaprasad, T., Michael Kifer, and David Warren. (1989). On the Declarative Semantics of Inheritance Networks. *Proceedings of the 11th International Joint Conference on Artificial Intelligence*. Detroit, Michigan.

Pearl, Judea. (July, 1987). Probabilistic Semantics for Inheritance Hierarchies with Exceptions. *TR-93*. Cognitive Systems Laboratory, University of California, Los Angeles.

Pollard, Carl, and Ivan Sag. (1987). *Information-Based Syntax and Semantics, Volume 1: Fundamentals*. Centre for the Study of Language and Information, Stanford University, CA.

Popowich, Fred. (1989). Tree Unification Grammar. *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*. Vancouver, BC.

Popowich, Fred and Carl Vogel. (1990). Chart Parsing Head-Driven Phrase Structure Grammar. *CSS-IS TR 90-01, CMPT TR 90-01*. Simon Fraser University, Burnaby, BC.

Selman, Bart, and Hector Levesque. (1989). The Tractability of Path-Based Inheritance. *Proceedings of the 11th International Joint Conference on Artificial Intelligence*. Detroit, Michigan.

Shieber, Stuart. (1986). *An Introduction to Unification-Based Approaches to Grammar*. The University of Chicago Press, Chicago, IL.

Touretzky, David. (1986). *The Mathematics of Inheritance Systems*. Los Altos, CA: Morgan Kaufman.

Touretzky, David, John Horty, Richard Thomason. (1987). A Clash of Intuitions: The Current State of Nonmonotonic Inheritance Systems. *Proceedings of the 10th International Joint Conference on Artificial Intelligence*. Milan, Italy.

## Centre for Systems Science

---

• Simon Fraser University established the **Centre for Systems Science** as a provincial centre for advanced study. The objective of CSS is twofold: coordinate and combine advanced systems research at SFU, and encourage application of research results within industry in order to further the technological development of British Columbia.

• CSS has assembled three research groups to streamline its efforts. The orientation of these groups is as follows: intelligent systems, microelectronics, and computer and communications systems. Because of shared research interests, this latter group merges with LCCR (**Laboratory for Computer and Communications Research**) for the publication of technical reports.

• CSS's three groups involve over 80 skilled faculty researchers principally from SFU's Schools of Engineering Science and Computing Science, but also draw substantially on Mathematics, Physics, Geography, Philosophy, Linguistics, Psychology, Communication, and Kinesiology.